Java Application Server Guide

For Version 10.3 or Later

Developer

Contents

Introduction to Java Application Server Guide 5
Organization of This Document 5
See Also 6

Application Server Overview 7
JBoss in OS X Server 7
Three Deployment Configurations 8

Configuring Applications 9
Starting the Application Server 9
Configuring Your Application 11
- Starting the JBoss Deployment Tool 11
- Loading Your Application 11
- Configuring Your Application's Components 14
- Saving a Configured Application 16

Deploying Your Application 16

Configuring and Deploying Sun's Pet Store 17
Set Up the Environment 17
Configure the Pet Store Enterprise Application 18
- Open the petstore.ear File in the Deployment Tool 18
- Configure Application-Wide Settings 19
- Configure the AsyncSender Enterprise Bean 20
- Configure the Catalog Enterprise Bean 20
- Configure the Customer Module 20
- Configure the PetStore Web Application 24
- Configure the SignOn Module 25
- Configure the User Enterprise Bean 25
- Configure the Counter Enterprise Bean 25
- Save the Application 25

Configure the Supplier Enterprise Application 26
- Open the supplier.ear file in the Deployment Tool 26
- Configure Application-Wide Settings 26
- Configure the Supplier Module 26

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Configure the SupplierPurchaseOrder Module 27
Configure the Supplier Web Application Module 29
Save the Application 29
Deploy and Test the Application 29

Administering Application Servers 31
Logging In to the Management Tool 31
Choosing a Task 33
Managing Application Servers 34
Configuring Application Servers 37
Monitoring Application Servers 38
Starting and Stopping Services 40
Creating a Data Source 40
Creating a Topic or a Queue 41
Deploying Applications 41

Balancing User Load and Replicating Sessions 42
Distributable Applications 42
Load Balancing and Clustering 43
Enable Load Balancing in the Web Server 43
Enable Load Balancing in the Application Servers 46
Test the Configuration 47

Document Revision History 48

Glossary 49

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Figures and Tables

Configuring Applications 9

Figure 2-1 The Load Application window of the deployment tool 12
Figure 2-2 The Loaded Application window 12
Figure 2-3 The deployment-tool main window 13
Figure 2-4 The Quick Config pane of a component's configuration window 14
Figure 2-5 A module-settings window 15
Figure 2-6 The Save Application window of the deployment tool 16

Configuring and Deploying Sun's Pet Store 17

Table 3-1 Relationship information for the Customer module 22

Administering Application Servers 31

Figure 4-1 The Configuration window of the management tool 34
Figure 4-2 The JBoss Management Console window 34
Figure 4-3 The JBoss Management Console window showing the Security Configuration pane of the log-in configuration service 35
Figure 4-4 The JBoss Management Console window showing the Statistics pane of the Pet Store ShoppingCart enterprise bean 36
Figure 4-5 The JBoss Management Console window showing one of the configuration panes for the JMS Directory Service 38
Figure 4-6 The JBoss Management Console window showing the statistics of the Deploy Service 39
Figure 4-7 The JBoss Management Console window showing the Start or Stop Services pane 40

Balancing User Load and Replicating Sessions 42

Figure 5-1 The WebApp window of the `petstore.ear` archive 43
Figure 5-2 The `workers.properties` file in `/etc/httpd` 45

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Introduction to Java Application Server Guide

Note: This document was previously titled *OS X Server Java Application Server Administration*.

JBoss is a powerful Java-based open-source application server that is very popular among Java 2, Enterprise Edition (J2EE) application developers. This document describes how to configure and use the JBoss application server in OS X Server, which lets you deploy J2EE applications easily and reliably.

This document is intended for system administrators, J2EE application assemblers, and developers. It assumes you have a solid background in system administration and J2EE technology. You must be familiar with OS X Server, especially how to use Terminal to issue shell commands. Knowledge of database engines, such as MySQL, is helpful but not required.

Organization of This Document

This document has the following chapters:

- Application Server Overview (page 7) provides an overview of JBoss for OS X Server.
- Configuring Applications (page 9) explains how to perform certain tasks with the deployment tool, such as opening, configuring, and saving application archives.
- Configuring and Deploying Sun's Pet Store (page 17) walks you through configuring and deploying Sun's Pet Store application in OS X Server.
- Administering Application Servers (page 31) teaches you how to manage application servers, which are JBoss instances running on one or more computers.
- Balancing User Load and Replicating Sessions (page 42) explains how to enable an application to be distributable among cluster nodes and walks you through configuring HTTP load balancing for Pet Store using three computers.

This document also contains a revision history, a glossary, and an index.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

See Also

To use this document to its fullest, you should download its companion files, which are packaged in `Application_Server_companion.zip`, located in the same webpage from which you obtained this document.

For an introduction to J2EE, visit http://java.sun.com/j2ee. You can get detailed information on JBoss at http://jboss.org.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Application Server Overview

JBoss is an open-source highly popular Java-based application server. Based on the Java 2, Enterprise Edition (J2EE) platform, JBoss provides an affordable delivery system for enterprise applications. Applications that follow the J2EE standard can be deployed on other application servers, such as WebLogic, WebSphere, and JRun, with little or no modification. JBoss provides many useful features in addition those defined in the J2EE standard, including support for clustering, session replication, mail, and security.

OS X Server includes two easy-to-use, HTML-based tools that facilitate the configuration of J2EE applications for deployment: The deployment tool and the management tool. The deployment tool allows you to open application or component archives (EAR files, WAR files, JAR files, SAR files, and so on) without having to manually decompress the archives. The application lets you view or change the values of settings specified in the `META-INF` and `WEB-INF` directories of the archives. See Configuring Applications (page 9) for more information.

The management tool allows you to manage application servers (JBoss instances) running on one or more computers. This application lets you start and stop services provided by individual application servers, configure services, and create data sources, queues, and topics. See Administering Application Servers (page 31) for details.

This chapter provides an overview of JBoss for OS X Server.

JBoss in OS X Server

OS X Server version 10.3 includes JBoss version 3.2.2RC2. To provide a high level of availability, OS X Server includes a "watchdog" process that ensures that the application server is always running (if you turn on the application server in Server Admin). If the application server freezes or crashes, the daemon restarts it automatically.

In addition, OS X Server offers load balancing and session failover through Apache and JBoss:

- Apache, coupled with the `mod_jk` plug-in, provides HTTP load balancing with session affinity (sticky sessions) and connects to JBoss instances through AJP connectors.
- JBoss offers session failover through HTTP session state replication in the cluster configuration.
- JBoss also provides load balancing for enterprise beans, including failover for stateful session beans, and support for session affinity.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

In OS X Server, JBoss is configured to use Tomcat (using the AJP connector) as its web server and servlet container. In addition, HTTP and HTTPS (through port 8443) are enabled by default.

You can manage the application server from the Server Admin application. This provides you with a simple way to start, stop, and monitor the application server. You can use the command line, if you prefer.

OS X Server includes two applications that allow you to deploy applications on JBoss and monitor their performance. They are the JBoss deployment tool and the JBoss management tool.

The deployment tool allows you to configure an application or an application component so that, for example, it accesses the appropriate data sources and database tables when it's run. This is how application developers decouple business logic from the database engine that is used to persist data. That way, you can use the database engine that meets your needs and not the one the developer used while developing the application. For details on the deployment tool, see Configuring Applications (page 9).

The management tool lets you administer the local (running on the computer you are logged in to) application server, and monitor local and remote (running on a computer in the local network) application servers. As part of administering an application server, you may start and stop services, configure services, deploy applications, and add data sources, queues, and topics. When monitoring an application server, the management tool lets you access the statistics provided by the resources and services running on it. For example, a service may indicate its name, its purpose, and when it was started. For more information on the management tool, see Administering Application Servers (page 31).

Three Deployment Configurations

In OS X Server, all the JBoss configurable settings are set up for maximum J2EE compliance. There are three standard deployment configurations in JBoss for OS X Server:

- The development configuration offers increased logging and also consults schema documents. As a result, an application is not deployed when the configuration files do not adhere to their respective schemas.
- The standalone configuration is set up for high performance on a single server.
- The cluster configuration is optimized for high performance on a cluster of servers. This includes load balancing as well as session replication among stateful session beans and HTTP sessions.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Configuring Applications

Before you can deploy an application on an application server, you have to start the application server and then configure or *assemble* the application. This is the process through which you specify data sources, database mappings, JNDI resources, and so on.

You configure J2EE applications by modifying XML files in `META-INF` and `WEB-INF` directories in application archives. Performing this task manually is tedious and error prone. The JBoss deployment tool allows you to configure applications without having to unarchive EAR files, WAR files, or JAR files, as the tool lets you configure these files directly.

This chapter explains how to start the application server and configure and deploy your application.

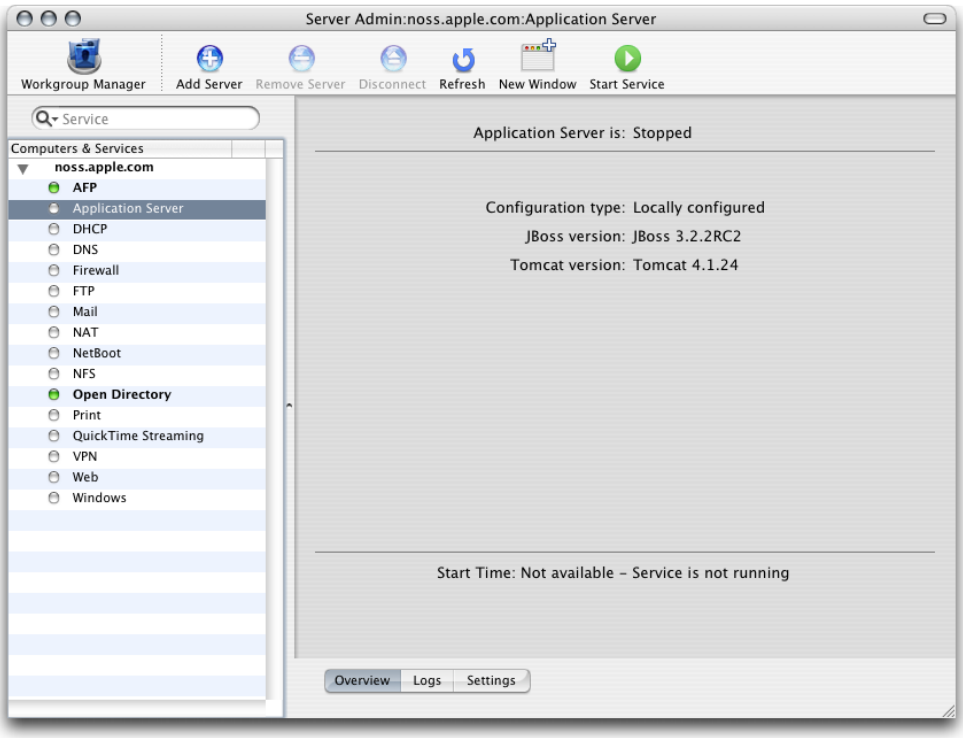
Starting the Application Server

To configure an application using the deployment tool, you must connect to a running application server. Follow these steps to start the application server on a computer.

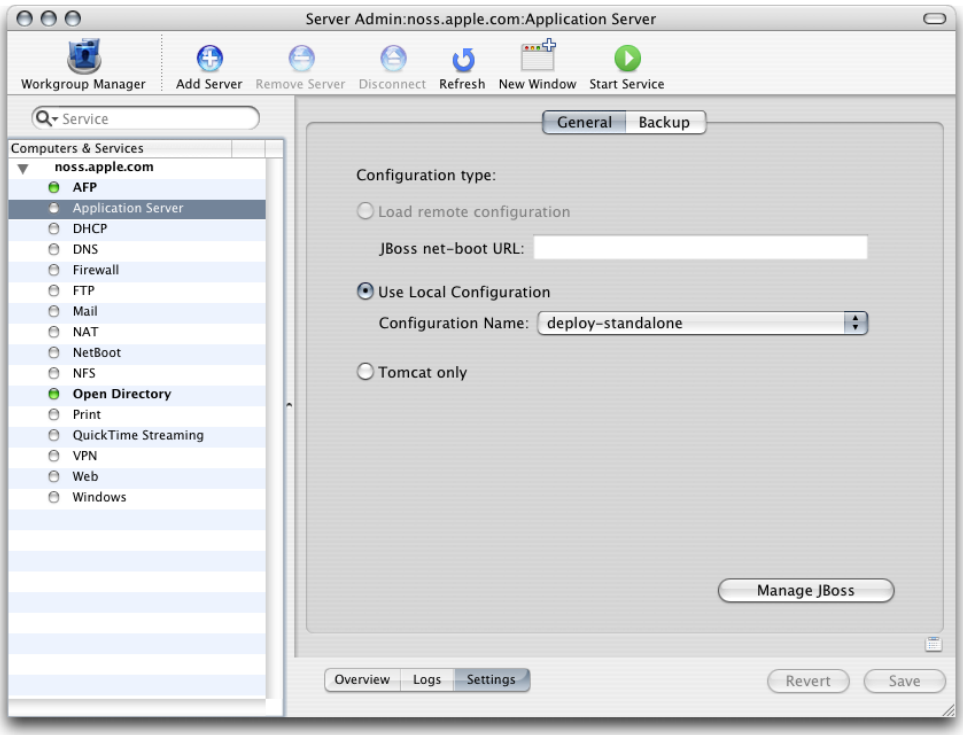
1. Launch Server Admin, located in `/Applications/Server`.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

2. In the Computers & Services list, select Application Server.



3. In the configuration pane, click Settings. From Configuration Name pop-up menu, choose the appropriate configuration.



Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

4. Click the Start Service toolbar button. After a few seconds the application server should be running. You can confirm that JBoss is running by accessing `http://localhost:8080` in your web browser. You should see a webpage titled Welcome to JBoss/Tomcat.

You can also start JBoss in Terminal with the following command:

```
$ /Library/JBoss/3.2/bin/run.sh -c deploy-standalone
```

To get detailed information on JBoss activities, use the `develop` configuration. This is useful when you need to make sure JBoss notices when you deploy or undeploy a module, or when you need to determine whether exceptions are thrown as JBoss starts a deployed application. The `develop` configuration produces a detailed log of JBoss activities. It is more useful when you launch the application server from the command line because you see the results of actions immediately in the Terminal window from which you launch the application server.

Configuring Your Application

The following sections teach you how to start the deployment tool and configure your application.

Starting the JBoss Deployment Tool

To start the deployment tool, double-click `DeploymentTool.woa` in `/Library/JBoss/Applications` or enter the following command in Terminal:

```
$ /Library/JBoss/Applications/DeploymentTool.woa/DeploymentTool
```

After a moment, the Load Application window appears.

> **Note:** Running the JBoss Deployment Tool requires a web browser that supports Frames and Javascript. Some web browsers may need to have pop-up blocking disabled.

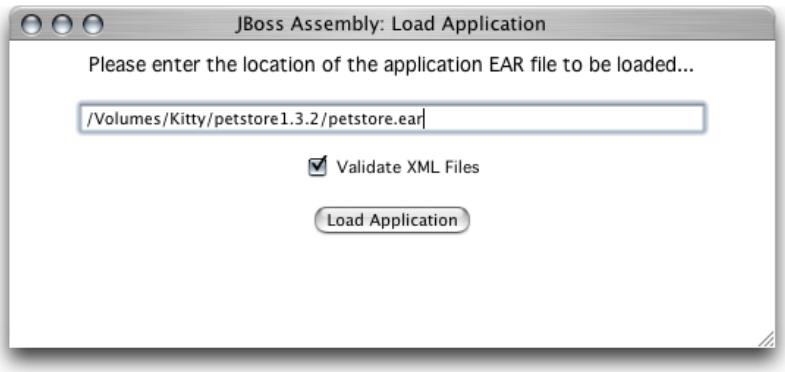
Loading Your Application

The Load Application window is where you specify the location of the application or component you want to configure. Although the window is titled Load Application, you can also use the deployment tool to configure EAR files, WAR files, and JAR files.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Figure 2-1 shows the Load Application window.

Figure 2-1 The Load Application window of the deployment tool


JBoss Assembly: Load Application
Please enter the location of the application EAR file to be loaded...
/Volumes/Kitty/petstore1.3.2/petstore.ear
Validate XML Files
Load Application

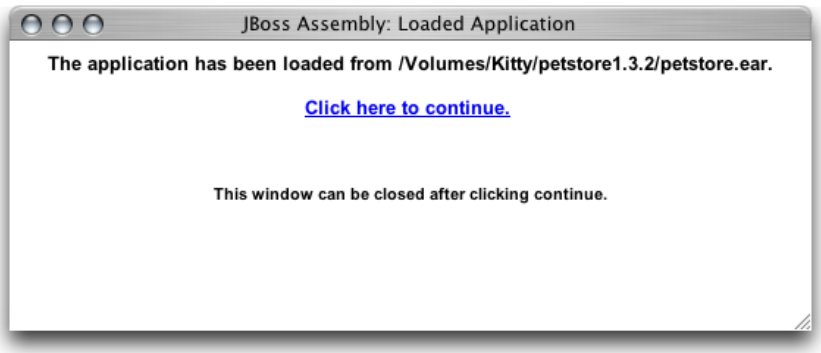
1. Enter the full path to the file in the text field in the Load Application window, and click Load Application.

   Note: The file path you enter in the text field is from the perspective of the server the deployment tool runs on. That is, if you access the deployment tool from a web browser that runs on a different computer, the archive you configure must reside on the server, not the computer the web browser runs on.

   Normally, you cannot save an application with invalid XML files. That is, you have to configure all the elements that show up in red in the main window. You can override this by deselecting Validate XML Files in the Load Application window. However, you may not be able to reload an application that has been saved in this state.

   After the deployment tool loads the application, it displays the Loaded Application window, shown in Figure 2-2.

Figure 2-2 The Loaded Application window


JBoss Assembly: Loaded Application
The application has been loaded from /Volumes/Kitty/petstore1.3.2/petstore.ear.
Click here to continue.
This window can be closed after clicking continue.

2. Click "Click here to continue" to move on.

   The deployment tool displays the main window (also known as the navigation window). The main window presents a hierarchy of components generated from the XML files present in the `META-INF` and `WEB-INF` directories of the components contained in the archive you opened. For example, Figure 2-3 shows the components present in the `petstore.ear` file of Sun Microsystem's Pet Store application. You must configure the items in red to save the application. Configuring Your Application's Components (page 14) shows you how to do this.

Figure 2-3 The deployment-tool main window

Figure 2-3 shows the components of the `petstore.ear` archive. The following list describes some of the items in the main window:

- **PetStoreEAR (Application)** Represents the Pet Store enterprise-application archive.
- **Application Settings** Clicking this link allows you to configure settings that affect all the modules in the archive when the application is deployed.

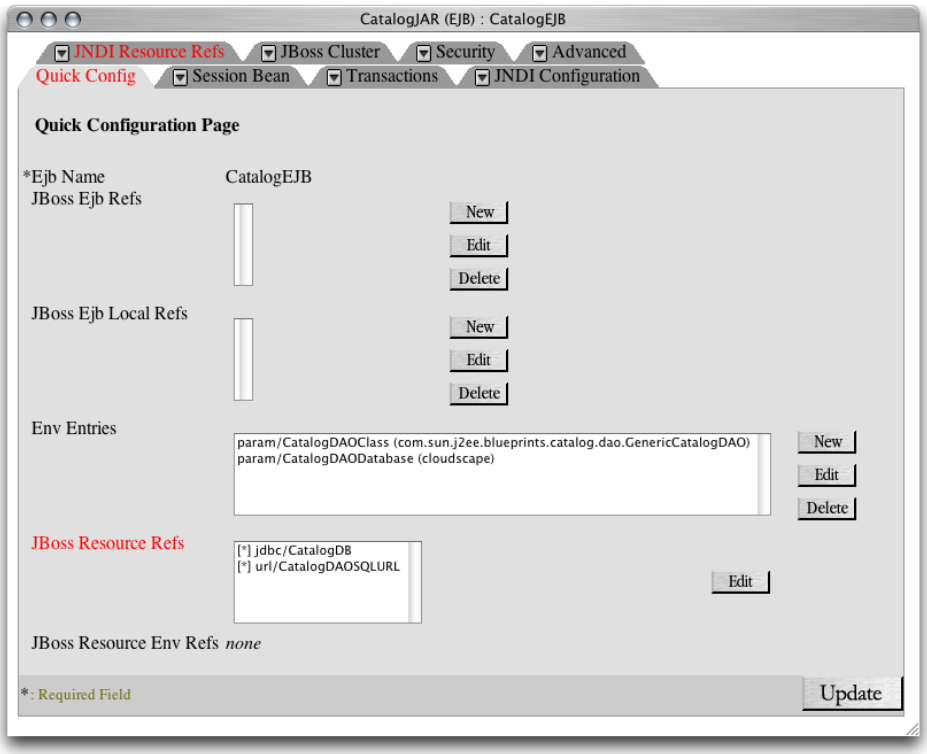
2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

- **AsyncSenderEJB (EJB)** Represents the archive (JAR file) that contains the files that define the AsyncSender enterprise bean (the `asyncsender-ejb.jar` file). Clicking the Module Settings link lets you configure module-wide settings and set default values for some settings for all the enterprise beans defined in the module. See Configure the Customer Module (page 20) for an example.
- **PetStoreWAR (WebApp)** Represents the archive (WAR file) that contains the files that define the web module of the Pet Store enterprise application.

Configuring Your Application's Components

To configure a component, you choose it from the main window by clicking the appropriate link. This causes the deployment tool to display the configuration window for the component. As you can see in Figure 2-4, this is a tabbed window that contains one or more panes, which you use to configure specific aspects of the component. The configuration window also contains a Quick Config pane, which contains elements of the component that you must configure for the application to be deployable. Figure 2-4 shows the Quick Config pane of the CatalogEJB module. It indicates that the JBoss resource references must be configured. The JBoss resource references also appear in the JNDI Resource Refs pane. However, you need to configure them in only one of the two panes.

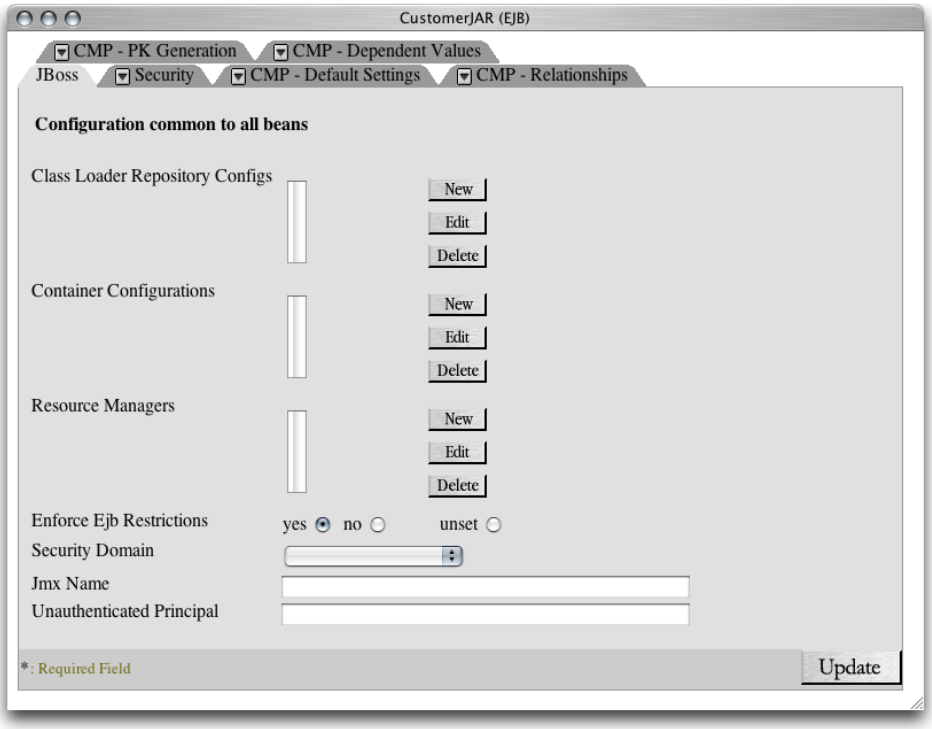
Figure 2-4 The Quick Config pane of a component's configuration window



> **Note:** For applications with many components, you may want to configure the Quick Config pages of the invalid components and save the application. Then you can configure each component further in a piecemeal fashion.

Some settings apply to an entire module, for example, security roles. In addition, some module settings serve as defaults for settings of individual components in the module. Figure 2-5 shows some of the module settings of the CustomerJAR module of the `petstore.ear` application. Configuring modules settings can help to speed up the configuration of a module. See Configure Module-Wide Settings (page 21) for an example.

Figure 2-5 A module-settings window



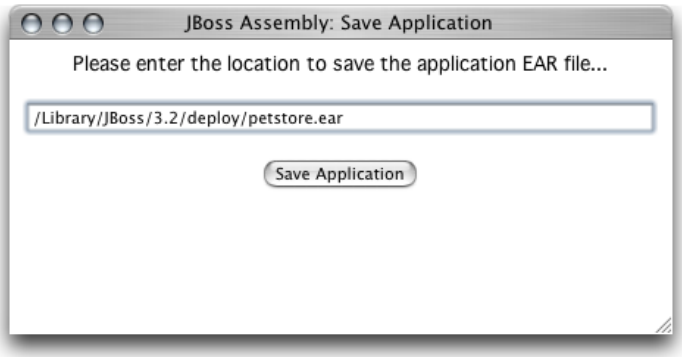
After configuring the elements in a pane, you click Update to confirm the changes. Keep in mind that the changes are not saved until you save the application.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Saving a Configured Application

To save a configured application, that is, when no components are shown in red in the main window, click Save in the top of the main window. The Save Application window (Figure 2-6) appears.

Figure 2-6 The Save Application window of the deployment tool


JBoss Assembly: Save Application
Please enter the location to save the application EAR file...
/Library/JBoss/3.2/deploy/petstore.ear
Save Application

Enter the destination of the configured application in the text field of the Save Application window, and click Save Application.

Deploying Your Application

To deploy a configured application from the deployment tool, simply save the application to `/Library/JBoss/3.2/deploy` in a single-server deployment or `/Library/JBoss/3.2/farm` in a cluster deployment.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Configuring and Deploying Sun's Pet Store

Pet Store is a sample J2EE application from Sun Microsystems. Pet Store showcases the power and flexibility of the J2EE platform. This chapter provides a tutorial on the configuration of Sun's Pet Store for deployment in OS X Server.

Sun's Pet Store is comprised of several applications. This tutorial uses the Pet Store enterprise application and the Supplier enterprise application.

In this tutorial you obtain the Pet Store files from Sun, prepare your OS X Server system for a Pet Store deployment, and configure the Pet Store and Supplier applications for deployment on JBoss.

> **Note:** The companion files of this document include configured Pet Store archives that are ready for deployment using MySQL as the database engine. They're at `Application_Server_companion/Configured_Pet_Store_archives`. You need to perform only the steps listed in Set Up the Environment (page 17) and Deploy and Test the Application (page 29).

See http://developer.java.sun.com/developer/releases/petstore for more information on Sun's Pet Store application.

Set Up the Environment

Before you can configure an enterprise application for deployment in OS X Server, make sure that you have all the files you need. Then create any necessary tables in your database, and ensure that the appropriate processes are running:

1. Get Pet Store from Sun.

   Download the Pet Store enterprise application from http://developer.java.sun.com/developer/releases/petstore, and place the `petstore1.3.2` directory in your home directory. (Pet Store 1.3.2 is also included as part of this document's companion files.)

2. Configure MySQL:

   a. Launch MySQL Manager, located in `/Applications/Server`.

   b. Click the lock button, and authenticate as the system administrator.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

c. Click Install and then click Start.

d. Quit MySQL Manager.

3. Create the Pet Store Tables:

   a. In Terminal, run the `mysql` command-line tool.

   b. Enter `use test` and press Return.

   c. Copy the text in `Application_Server_companion/Pet_Store_resources/create_tables_sql.txt` (in this document's companion files) to the Clipboard, and paste its contents into the `mysql` command-line tool.

   d. Enter `quit` and press Return.

4. Deploy the `petstore-destinations-service.xml` file.

   To prepare JBoss for running Pet Store, copy the `Application_Server_companion/Pet_Store_resources/petstore-destinations-service.xml` file included in this document's companion files to `/Library/JBoss/3.2/deploy`.

5. Start the application server.

   You must be running the application server to configure Pet Store. Make sure the application server is running on your computer. See Starting the Application Server (page 9) for details.

Configure the Pet Store Enterprise Application

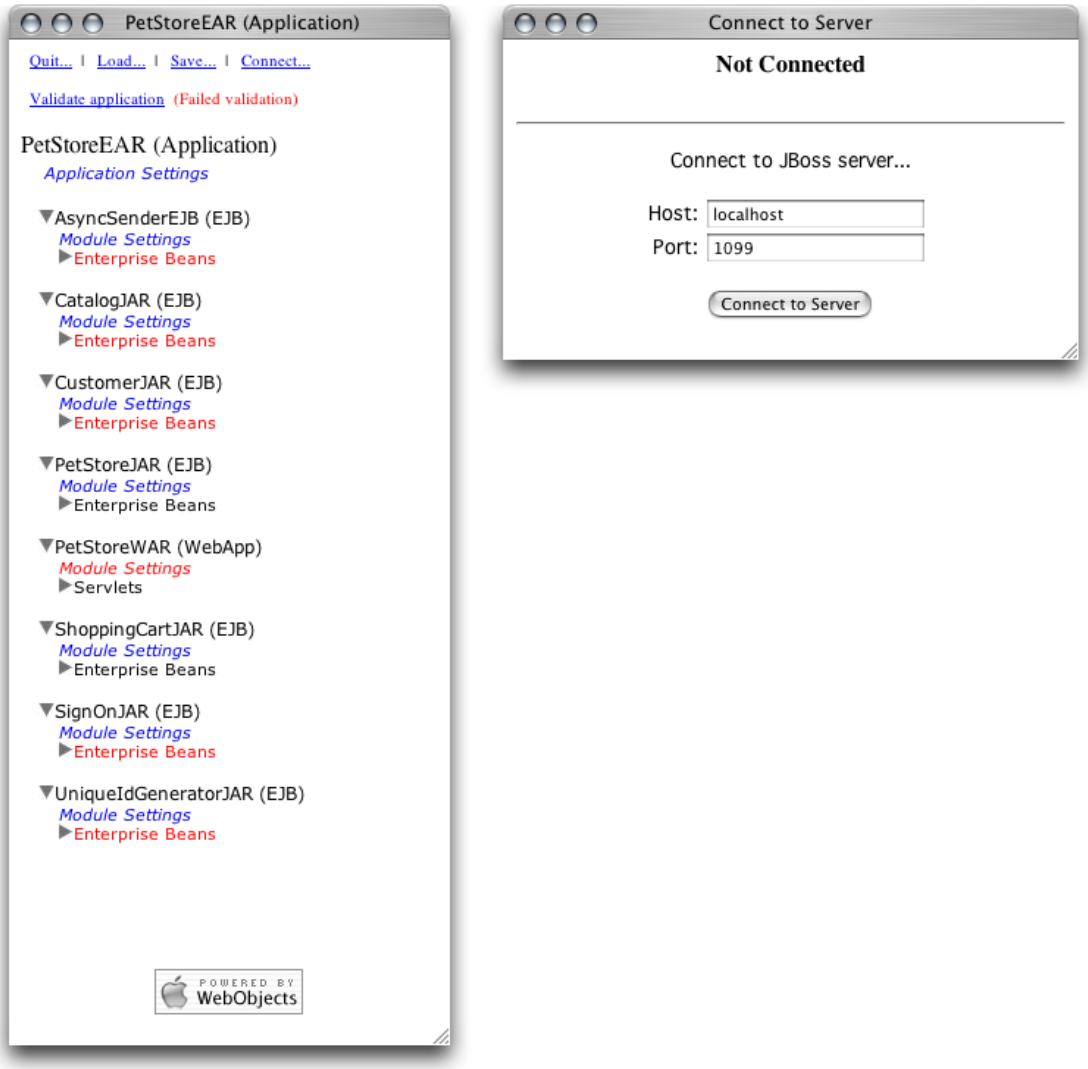
This section guides you through the steps required to configure the `petstore.ear` file so that it can be deployed in OS X Server. This process involves specifying the data source that enterprise beans use to obtain a connection to the database, mapping the enterprise beans' CMP fields to table columns, defining relationships among enterprise beans, and so on.

Open the petstore.ear File in the Deployment Tool

1. Launch the JBoss deployment tool. (See Starting the JBoss Deployment Tool (page 11) for details.)
2. In the Load Application window, enter the full path to the `petstore.ear` file in the text field, and click Load Application.
3. Click "Click here to continue."
4. In the PetStoreEAR (Application) window (also known as the main window or the navigation window), click Connect.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

5. In the Connect to Server window, enter the host name and the port of the application server. (By default the host is `localhost` and the port is `1099`.)



6. Click Connect to Server.
7. Close the Connect to Server window.

Configure Application-Wide Settings

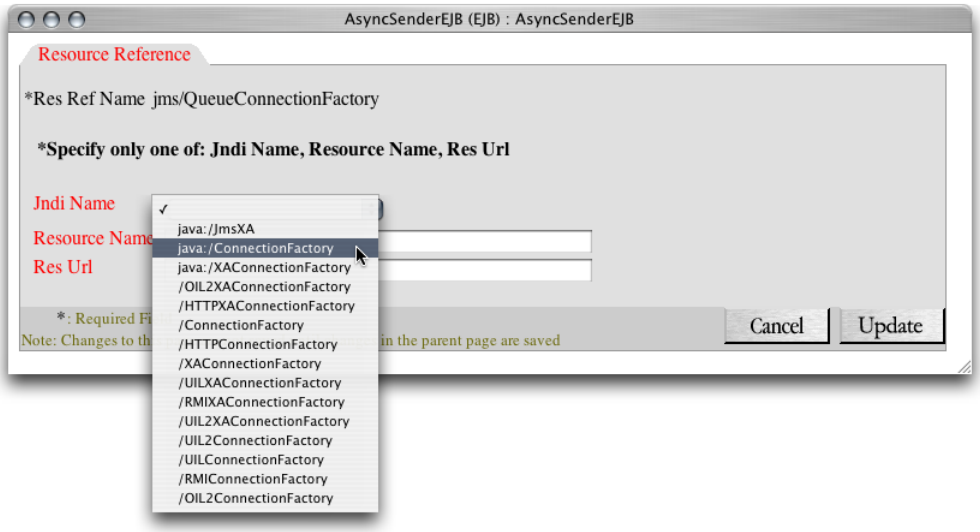
In this section you configure the settings that affect all the modules in the `petstore.ear` archive.

1. Under PetStoreEAR (Application), click Application Settings.
2. In the PetStoreEAR window, click the JBoss tab.
3. Set J2EE Compliant Class Loading to no, and click Update.
4. Close the window.

Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Configure the AsyncSender Enterprise Bean

1. Under AsyncSenderEJB (EJB), click Enterprise Beans. Then click AsyncSenderEJB.
2. In the AsyncSenderEJB window, select `jms/QueueConnectionFactory` in the JBoss Resource Refs list, and click Edit.
3. From the Jndi Name pop-up menu, choose `java:/ConnectionFactory` , and click Update.



4. In the JBoss Resource Env Refs list, select `jms/AsyncSenderQueue` , and click Edit.
5. From the Jndi Name pop-up menu, choose `/queue/supplier/PurchaseOrderQueue`, and click Update.
6. Click Update to finish configuring the AsyncSender enterprise bean, and close the window.

Configure the Catalog Enterprise Bean

1. Under CatalogJAR (EJB), click Enterprise Beans. Then click CatalogEJB.
2. In the JBoss Resource Refs list in the CatalogEJB window , select `jdbc/CatalogDB`, and click Edit.
3. From the Jndi Name pop-up menu, choose `java:MySqlDS` , and click Update.
4. In the JBoss Resources Refs list in the CatalogEJB window, select `url/CatalogDAOSQLURL` , and click Edit.
5. In the Resource URL text field, enter `http://localhost:8080/petstore/CatalogDAOSQL.xml`, and click Update.
6. Click Update to finish configuring the Catalog enterprise bean, and close the window.

Configure the Customer Module

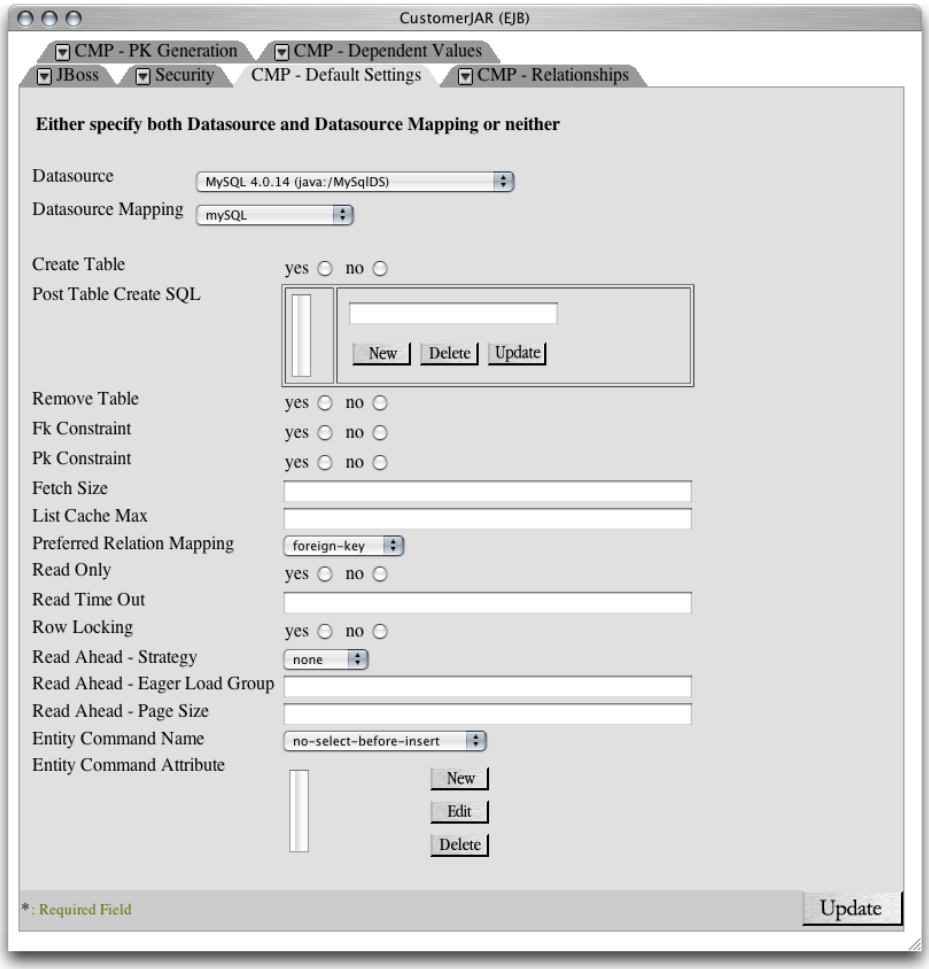
The following sections guide you through the configuration of the Customer module.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Configure Module-Wide Settings

This section walks you through setting the data source and data-source mapping for the enterprise beans in the Customer module. It also details how to configure the relationships between some of the enterprise beans.

1. Under CustomerJAR (EJB), click Module Settings.
2. In the CustomerJAR window, click the CMP - Default Settings tab.
3. From the Datasource pop-up menu, choose MySQL 4.0.14 (java:/MySqlDS).
4. From the Datasource Mapping pop-up menu, choose `mySQL`, and click Update.
5. From the Entity Command Name pop-up menu, choose `no-select-before-insert`.

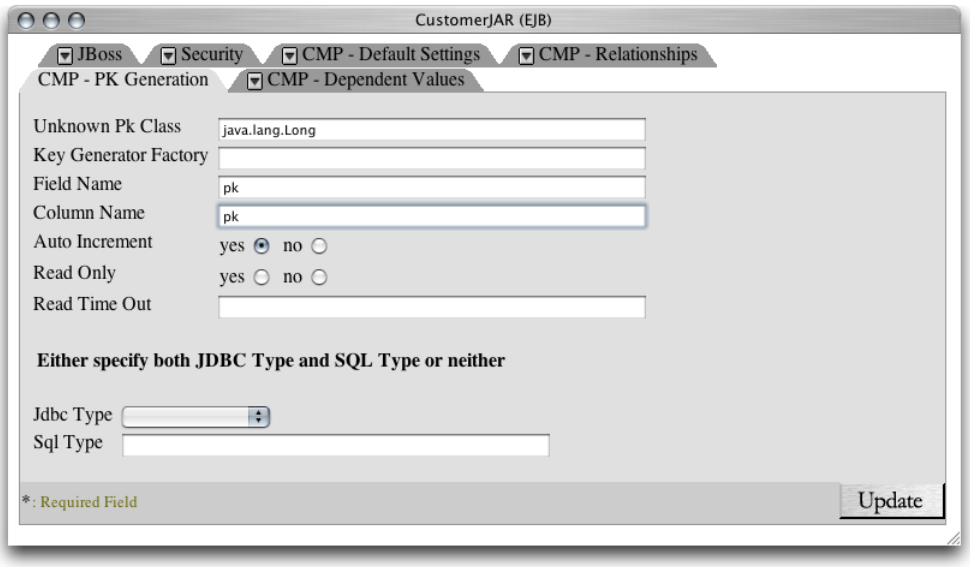


The advantage of configuring the database and the data-source mapping at the module level is that the settings apply to all the enterprise beans in the module. Therefore, you don't have to configure those settings for each enterprise bean in the module, unless they differ from the ones set for the module.

6. Click the CMP - PK Generation tab.
7. In the Unknown Pk Class text field, enter `java.lang.Long`.
8. In the Field Name text field, enter `pk`.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

9. In the Column Name text field, enter `pk`.
10. Set Auto Increment to yes.



11. Click the CMP - Relationships tab.
12. Configure the relationships.

    Table 3-1 lists the relationship information for the customer module.

    Table 3-1 Relationship information for the Customer module

Relationship name	Role name	Column name
CustomerEJB-AccountEJB Relationship	AccountEJB	`account_fk`
ContactInfoEJB-AddressEJB Relationship	AddressEJB	`address_fk`
CustomerEJB-ProfileEJB Relationship	ProfileEJB	`profile_fk`
AccountEJB-ContactInfoEJB Relationship	ContactInfoEJB	`contactInfo_fk`
AccountEJB-CreditCardEJB Relationship	CreditCardEJB	`creditCard_fk`

    Perform the following steps to configure each relationship listed in Table 3-1.

    a. In the Ejb Relations list, select the relationship to configure, and click Edit.
    b. In the Relationship Roles list, click the corresponding relationship role.
    c. Click New next to the Key Fields list.
    d. In the Field Name text field, enter `pk`.
    e. From the Column Name pop-up menu, choose the name of the appropriate column, and click Update.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

f. Click Update to finish configuring the relationship role.

g. Click Update to finish configuring the relationship.

13. Click Update to finish configuring the Customer module settings, and close the window.

Configure the Account Enterprise Bean

1. Under CustomerJAR (EJB), click Enterprise Beans. Then click AccountEJB.
2. From the Table Name pop-up menu in the AccountEJB window, choose `PS_Account`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click the CMP - Mapping tab. From the Entity Command Name pop-up menu, choose `mysql-get-generated-keys`.
5. Click Update to finish configuring the Account enterprise bean, and close the window.

Configure the Address Enterprise Bean

1. Under Enterprise Beans under CustomerJAR (EJB), click AddressEJB.
2. From the Table Name pop-up menu in the AddressEJB window, choose `PS_Address`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click the CMP - Mapping tab. From the Entity Command Name pop-up menu, choose `mysql-get-generated-keys`.
5. Click Update to finish configuring the Address enterprise bean, and close the window.

Configure the ContactInfo Enterprise Bean

1. Under Enterprise Beans under CustomerJAR (EJB), click ContactInfoEJB.
2. From the Table Name pop-up menu in the ContactInfoEJB window, choose `PS_ContactInfo`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click the CMP - Mapping tab. From the Entity Command Name pop-up menu, choose `mysql-get-generated-keys`.
5. Click Update to finish configuring the ContactInfo enterprise bean, and close the window.

Configure the CreditCard Enterprise Bean

1. Under Enterprise Beans under CustomerJAR (EJB), click CreditCardEJB.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

2. From the Table Name pop-up menu in the CreditCardEJB window, choose `PS_CreditCard`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click the CMP - Mapping tab. From the Entity Command Name pop-up menu, choose `mysql_get_generated_keys`.
5. Click Update to finish configuring the CreditCard enterprise bean, and close the window.

Configure the Customer Enterprise Bean

1. Under Enterprise Beans under CustomerJAR (EJB), click CustomerEJB.
2. From the Table Name pop-up menu in the CustomerEJB window, choose `PS_Customer`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click Update to finish configuring the Customer enterprise bean, and close the window.

Configure the Profile Enterprise Bean

1. Under Enterprise Beans under CustomerJAR (EJB), click ProfileEJB.
2. From the Table Name pop-up menu in the ProfileEJB window, choose `PS_Profile`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click the CMP - Mapping tab. From the Entity Command Name pop-up menu, choose `mysql_get_generated_keys`.
5. Click Update to finish configuring the Profile enterprise bean, and close the window.

Configure the PetStore Web Application

1. Under PetStoreWAR (WebApp), click Module Settings.
2. In the JBoss Resource Refs list in the PetStoreWAR window, select `jdbc/CatalogDB`, and click Edit.
3. From the Jndi Name pop-up menu, choose `java:/MySqlDS`, and click Update.
4. In the JBoss Resource Refs list, select `url/CatalogDAOSQLURL`, and click Edit.
5. In the Resource URL text field, enter `http://localhost:8080/petstore/CatalogDAOSQL.xml`, and click Update.
6. Click Update to finish configuring the PetStore web application, and close the window.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Configure the SignOn Module

1. Under SignOnJAR (EJB), click Module Settings.
2. In the SignOnJAR window, click the CMP - Default Settings tab.
3. From the Entity Command Name pop-up menu, choose `no-select-before-insert`, and click Update.
4. Close the window.

Configure the User Enterprise Bean

1. Under Enterprise Beans under SignOnJAR (EJB), click UserEJB.
2. From the Datasource pop-up menu in the UserEJB window, choose MySQL 4.0.14 (java:/MySqlDS).
3. From the Datasource Mapping pop-up menu, choose `mySql`.
4. From the Table Name pop-up menu, choose `PS_User`.
5. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
6. Click Update to finish configuring the User enterprise bean, and close the window.

Configure the Counter Enterprise Bean

1. Under UniqueIdGeneratorJAR (EJB), click Enterprise Beans. Then click CounterEJB.
2. From the Datasource pop-up menu in the CounterEJB window, choose MySQL 4.0.14 (java:/MySqlDS).
3. From the Datasource Mapping pop-up menu, choose `mySql`.
4. From the Table Name pop-up menu, choose `PS_Counter`.
5. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
6. Click Update to finish configuring the Counter enterprise bean, and close the window.

Save the Application

To save the configured PetStore application, click Save in the navigation window and choose a location for it:

1. Using the Finder or Terminal, create a directory under `/Library` named Configured_Apps.
2. In the main window, click Save.
3. In the text field in the Save Application window, enter `/Library/Configured_Apps/petstore.ear`, and click Save Application.
4. Close the window.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Configure the Supplier Enterprise Application

The following sections guide you through configuring the Supplier enterprise application.

Open the supplier.ear file in the Deployment Tool

1. In the PetStoreEAR window, click Load.
2. In the text field in the Load Application window, enter the path to the `supplier.ear` file, and click Load Application or press Return.
3. In the navigation window, click Connect.
4. If the Connect to Server window indicates that you're not connected to the application server, click Connect to Server.
5. Close the Connect to Server window.

Configure Application-Wide Settings

1. Under SupplierEAR (Application), click Application Settings.
2. In the SupplierEAR window, click the JBoss tab.
3. Set J2EE Compliant Class Loading to no, and click Update.
4. Close the window.

Configure the Supplier Module

The following sections explain how to configure the SupplierJAR module.

Configure the Module Settings

1. Under SupplierJAR (EJB), click Module Settings.
2. In the SupplierJAR window, click the CMP - Default Settings tab.
3. From the Datasource pop-up menu, choose MySQL 4.0.14 (java:/MySqlDS).
4. From the Datasource Mapping pop-up menu, choose `mySql`, and click Update.
5. Close the window.

Configure the Inventory Enterprise Bean

1. Under SupplierJAR (EJB), click Enterprise Beans. Then click InventoryEJB.
2. From the Table Name pop-up menu, choose `SUPP_Inventory`.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click Update to finish configuring the Inventory enterprise bean, and close the window.

Configure the OrderFulfillmentFacade Enterprise Bean

1. Under Enterprise Beans under SupplierJAR (EJB), click OrderFulfillmentFacade.
2. In the JBoss Resource Refs list in the OrderFulfillmentFacadeEJB window, select `url/EntityCatalogURL`, and click Edit.
3. In the Res URL text field, enter `http://localhost:8080/opc/EntityCatalog.jsp`, and click Update.
4. Click Update to finish configuring the OrderFulfillmentFacade enterprise bean, and close the window.

Configure the SupplierOrder Message-Driven Bean

1. Under Enterprise Beans under SupplierJAR (EJB), click SupplierOrderMDB.
2. In the JBoss Resource Refs list, select `jms/QueueConnectionFactory`, and click Edit.
3. From the Jndi Name pop-up menu, choose `/ConnectionFactory`, and click Update.
4. In the JBoss Resource Refs list, select `jms/TopicConnectionFactory`, and click Edit.
5. From the Jndi Name pop-up menu, choose `/ConnectionFactory`, and click Update.
6. In the JBoss Resource Env Refs list, select `jms/opc/InvoiceTopic`, and click Edit.
7. From the Jndi Name pop-up menu, choose `/topic/opc/InvoiceTopic`, and click Update.
8. Click Update to finish configuring the SupplierOrder message-driven bean, and close the window.

Configure the SupplierPurchaseOrder Module

These sections explain how to configure the SupplierPurchaseOrderJAR module.

Configure the Module Settings

1. Under SupplierPurchaseOrderJAR (EJB), click Module Settings.
2. In the SupplierPurchaseOrderJAR window, click the CMP - Default Settings tab.
3. From the Datasource pop-up menu, choose MySQL 4.0.14 (java:/MySqlDS).
4. From the Datasource mapping pop-up menu, choose `mySql`.
5. From the Entity Command Name pop-up menu, choose `no-select-before-insert`.
6. Click the CMP - PK Generation tab.
7. In the Unknown Pk Class text field, enter `java.lang.Long`.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

8. In the Field Name text field, Enter `pk`.
9. In the Column Name text field, enter `pk`.
10. Set Auto Increment to yes, and click Update.
11. Close the window.

Configure the Address Enterprise Bean

1. Under SupplierPurchaseOrderJAR (EJB), click Enterprise Beans. Then click AddressEJB.
2. From the Table Name pop-up menu in the AddressEJB window, choose `PS_Address`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click the CMP - Mapping tab. From the Entity Command Name pop-up menu, choose `mysql-get-generated-keys`.
5. Click Update to finish configuring the Address enterprise bean, and close the AddressEJB window.

Configure the ContactInfo Enterprise Bean

1. Under Enterprise Beans under SupplierPurchaseOrderJAR (EJB), click ContactInfoEJB.
2. From the Table Name pop-up menu in the ContactInfoEJB window, choose `PS_ContactInfo`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click the CMP - Mapping tab. From the Entity Command Name pop-up menu, choose `mysql-get-generated-keys`.
5. Click Update to finish configuring the ContactInfo enterprise bean, and close the window.

Configure the LineItem Enterprise Bean

1. Under Enterprise Beans under SupplierPurchaseOrderJAR (EJB), click LineItemEJB.
2. From the Table Name pop-up menu in the LineItemEJB window, choose `SUPP_LineItem`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click the CMP - Mapping tab. From the Entity Command Name pop-up menu, choose `mysql-get-generated-keys`.
5. Click Update to finish configuring the LineItem enterprise bean, and close the window.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Configure the SupplierOrder Enterprise Bean

1. Under Enterprise Beans under SupplierPurchaseOrderJAR (EJB), click SupplierOrderEJB.
2. From the Table Name pop-up menu in the SupplierOrderEJB window, choose `SUPP_SupplierOrder`.
3. Map the CMP fields to the appropriate column names by selecting the field in the Cmp Fields list, clicking Edit, choosing the corresponding column name from the Column Name list, and clicking Update.
4. Click the CMP - Mapping tab. From the Entity Command Name pop-up menu, choose `mysql-get-generated-keys`.
5. Click Update to finish configuring the SupplierOrder enterprise bean, and close the window.

Configure the Supplier Web Application Module

1. Under SupplierWAR (WebApp), click Module Settings.
2. In the JBoss Resource Env Refs list, select `jms/opc/InvoiceTopic`, and click Edit.
3. From the Jndi Name pop-up menu, choose `/topic/opc/InvoiceTopic`, and click Update.
4. In the JBoss Resource Refs list, select `jms/TopicConnectionFactory`, and click Edit.
5. From the Jndi Name pop-up menu, choose `/ConnectionFactory`, and click Update.
6. Click Update to finish configuring the Supplier web-application module, and close the window.

Save the Application

1. In the navigation window, click Save.
2. In the text field in the Save Application window, enter `/Library/Configured_Apps/supplier.ear`, and click Save Application.
3. Close the window.

Deploy and Test the Application

To deploy Pet Store in OS X Server, copy the configured files to `/Library/JBoss/3.2/deploy`. (You can also use the management tool to deploy the application. See Deploying Applications (page 41) for details.) After about a minute, open `http://localhost:8080/petstore` in your web browser. You could also have saved the EAR files directly to the JBoss `deploy` directory. However, it's generally safer to configure application files of undeployed archives.

Follow these steps to test Pet Store:

1. Open `http://localhost:8080/petstore` in a web-browser window.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

2. Click the link that takes you to the store.
3. In the Welcome to the BluePrints Petstore webpage, click Birds.
4. In the Items webpage, click Amazon Parrot.
5. In the Product webpage, click Add to Cart.
6. In the Cart webpage, click Check Out.
7. In the Sign On webpage, click Sign In.
8. In the Enter Order Information webpage, click Submit.

If you get an error page during the test, make sure JBoss is running and recheck the configuration settings described in Configure the Pet Store Enterprise Application (page 18) and Configure the Supplier Enterprise Application (page 26).

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Administering Application Servers

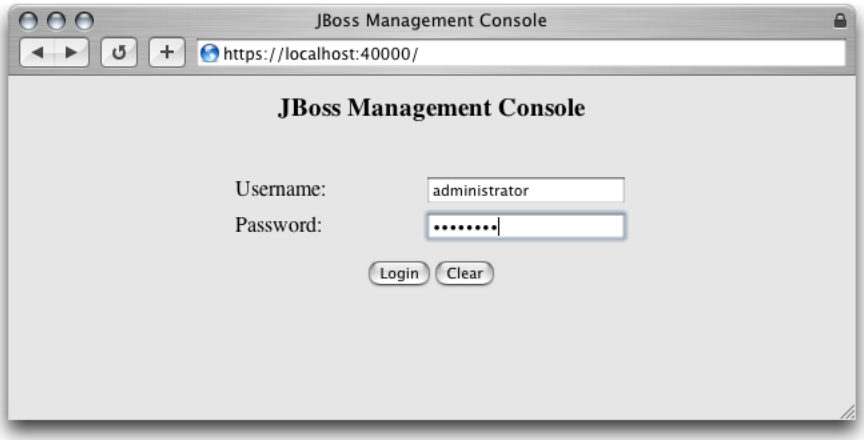
Application-server management involves configuring the services available in an application server, turning services on and off, deploying applications, and monitoring application-server resources. This chapter teaches how to manage application servers, which are JBoss instances running on one or more computers.

Logging In to the Management Tool

Before you can log in to the management tool, the tool must be running. You can launch the management tool by executing the following command:

```
$ /Library/JBoss/Applications/JBossManagement.woa/JBossManagement
```

You log in to the management tool through your web browser. To connect to the management tool, enter the following URL into the Address text field in your web browser: `https://localhost:40000`.



Alternatively, you can click Manage JBoss in the Settings pane in the Server Admin window.

To log in to the management tool you must enter the user name and password of an administrator of your computer or a user who is a member of the `appserveradm` group or the `appserverusr` group.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

There are two types of users with the authority to use the management tool: administrator users (which are members of the `appserveradm` group) and monitor users (which are members of the `appserverusr` group). The user defined while configuring OS X Server for the first time is added to the `appserveradm` group.

> **Note:** The management tool allows only registered users to manage, configure, or monitor application servers. If you enter the name of a registered user in the Username text field but enter an incorrect password in the Password text field, you can only monitor the application server. Also, only members of `appserveradm` and `appserverusr` can launch the management tool.

To authorize another user to manage application servers with the management tool, execute the following command in the command line, and restart your computer:

```
$ sudo /usr/bin/dscl . –create Groups/appserveradm GroupMembership <user_ID>
```

If you need to authorize a user to only monitor application servers with the management tool, execute the following command, and restart the computer:

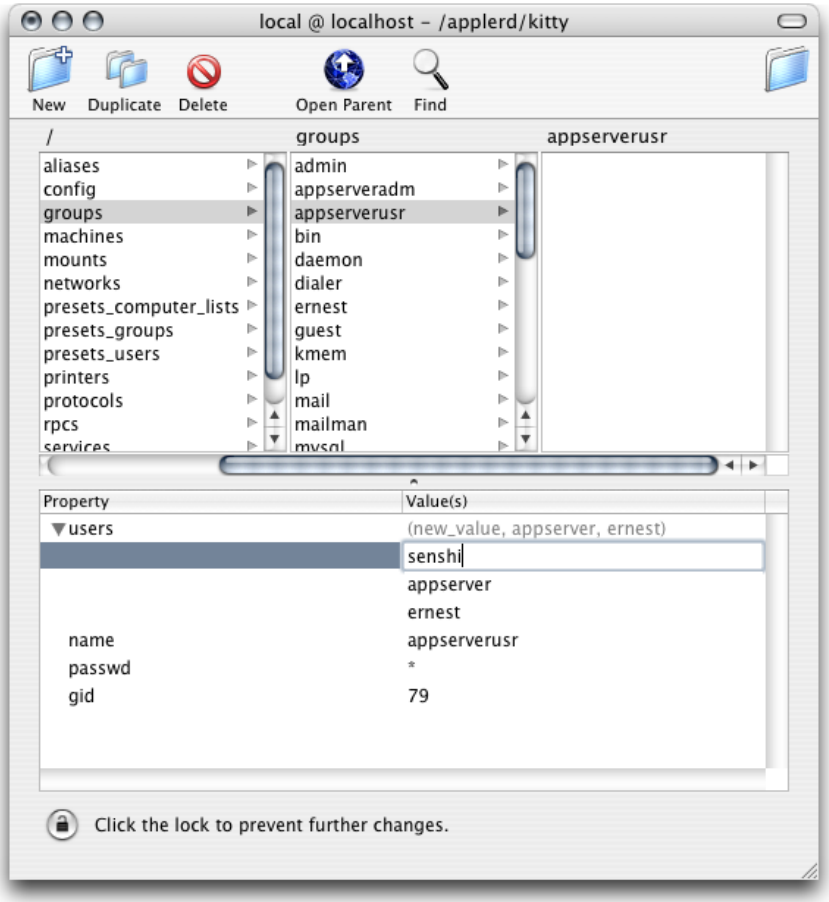
```
$ sudo /usr/bin/dscl . –create Groups/appserverusr GroupMembership <user_ID>
```

You can also use NetInfo Manager to add users to the `appserveradm` and `appserverusr` groups:

1. Launch NetInfo Manager, located in `/Applications/Utilities`.
2. If the lock in the bottom-left corner of the window is locked, click it and authenticate yourself and the server's administrator.
3. In the "/" column, select "groups." Then select the appropriate group in the "groups" column.
4. In the Properties list, select "users" and choose Directory > New Value.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

5. Replace `new_value` with the appropriate user name.



6. Save the changes and restart your computer.

Choosing a Task

After logging in to the management tool, the Configuration window appears. This window lets you choose the kind of activity you want to perform with the management tool. There are three activities to choose from:

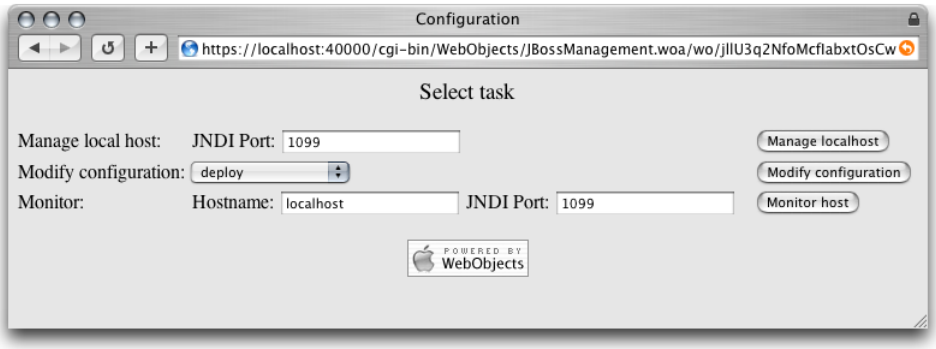
- **Managing**: Allows you to start and stop services, configure services, observe resource statistics, create data sources, create topics or queues, and deploy applications.
- **Configuring**: Lets you start and stop services, create data sources, create topics or queues, and deploy applications.
- **Monitoring**: Allows you to observe resource statistics of application servers.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Managing Application Servers

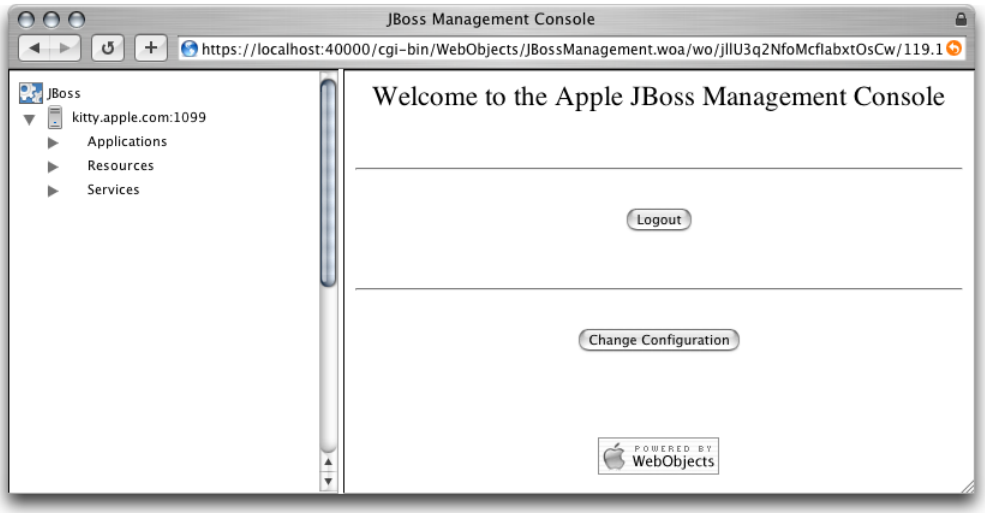
To manage an application server, enter the JNDI port of the application server you want to manage (by default, `1099`) in the Configuration window (shown in Figure 4-1), and click "Manage localhost".

Figure 4-1 The Configuration window of the management tool



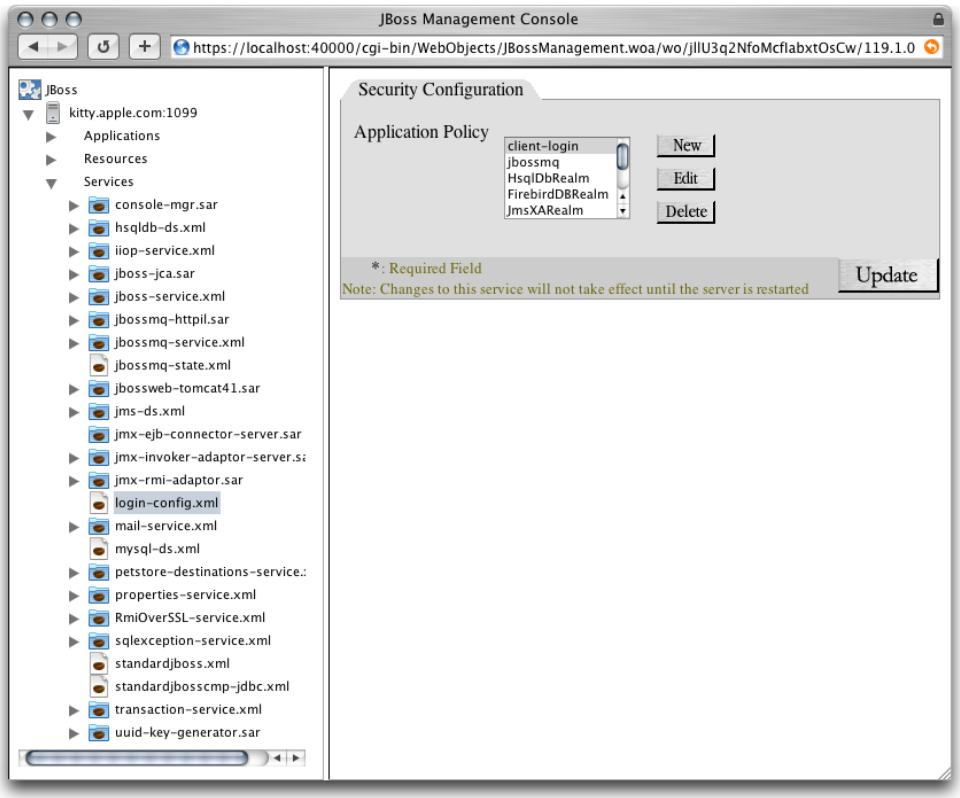
The left side of the JBoss Management Console window (shown in Figure 4-2) lists the application servers available and the resources they provide. You specify the resource you want to manage by clicking the triangle next to the appropriate resource type and selecting a resource from the list that appears.

Figure 4-2 The JBoss Management Console window



For example, to change an application server's security configuration, click the triangle next to the Services resource group and select `login-config.xml`. After that, the Security Configuration pane (shown in Figure 4-3) appears on the right side of the window, showing the application policy list, which you can modify by clicking the appropriate buttons. However, any changes you make take effect only after you restart the application server.

Figure 4-3 The JBoss Management Console window showing the Security Configuration pane of the log-in configuration service

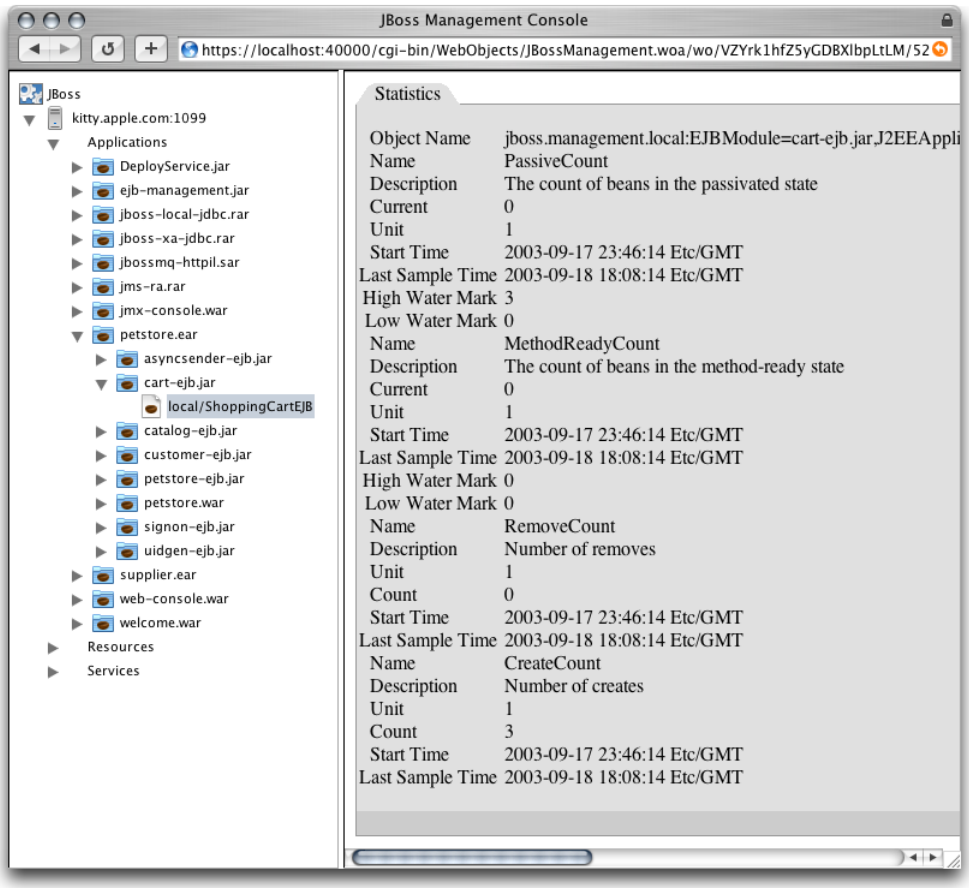


When you're managing the services deployed on the application server, you can also monitor the statistics of deployed applications and resources. For example, if you deploy Sun's Pet Store in your application server, log in to the management tool, choose to manage the application server from the Configuration window, and

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

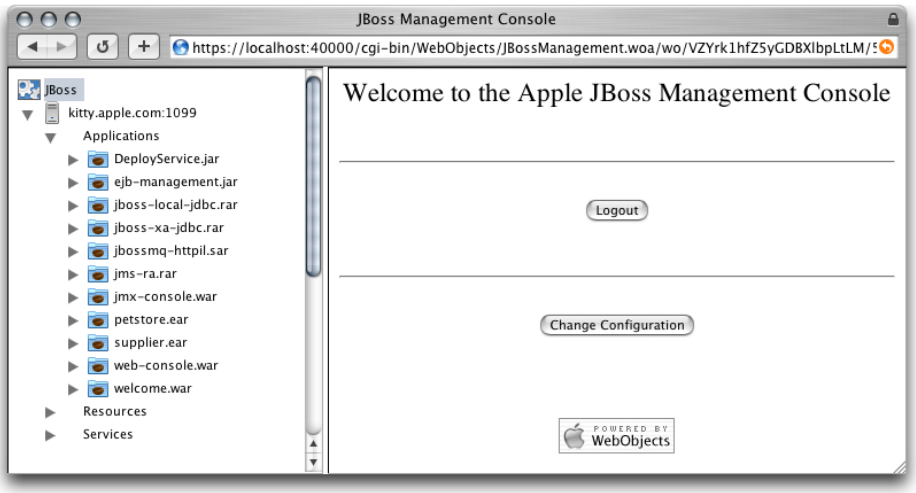
click `local/ShoppingCartEJB` under `cart-ejb.jar` under `petstore.ear` under the Applications group in the application-server list, the JBoss Management Console window displays the Statistics pane with information on the performance of the ShoppingCart enterprise bean, as shown in Figure 4-4.

Figure 4-4 The JBoss Management Console window showing the Statistics pane of the Pet Store ShoppingCart enterprise bean



2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

When you're done managing, click JBoss in the server list, and click Logout or Change Configuration.



Configuring Application Servers

To configure local application servers, choose the configuration you want to modify from the "Modify configuration" pop-up menu, and click "Modify configuration" in the Configuration page.

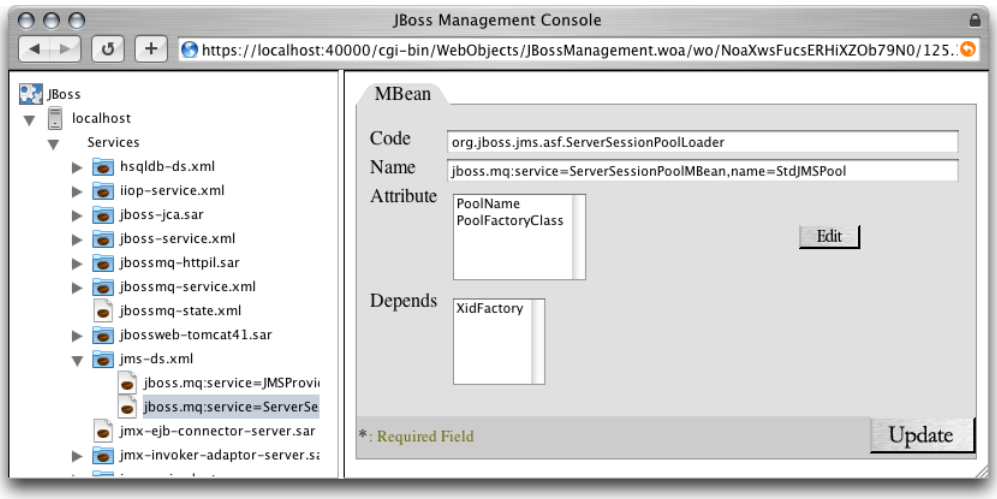
The JBoss Management Console window appears. This window allows you to select a service and change its configuration.

For example, to configure the transaction-connection factory service, select `jms-ds.xml` under services in the application-server list.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

You can also configure the provider and the session pool of the transaction-connection factory service by clicking the triangle next to `jms-ds.xml` in the application-server list and selecting the appropriate item, as shown in Figure 4-5.

Figure 4-5 The JBoss Management Console window showing one of the configuration panes for the JMS Directory Service



Note: After you're done making changes, you have to restart the application server for the changes to take effect.

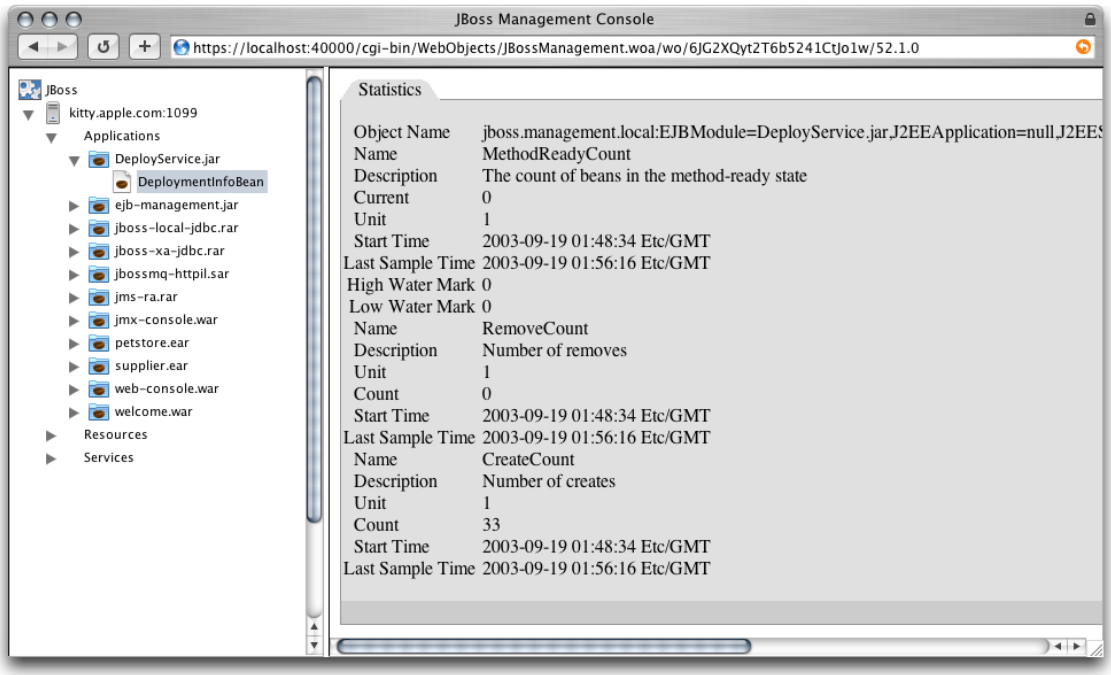
Monitoring Application Servers

To monitor application servers, enter the name of the computer on which the application server is running (by default, `localhost`) and the JNDI port of the application server (by default, `1099`) in the Configuration page, and click "Monitor host".

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

To view the statistics provided by particular applications, resources, or services, select the appropriate item in the application-server list. The statistics appear in the Statistics pane, shown in Figure 4-6.

Figure 4-6 The JBoss Management Console window showing the statistics of the Deploy Service

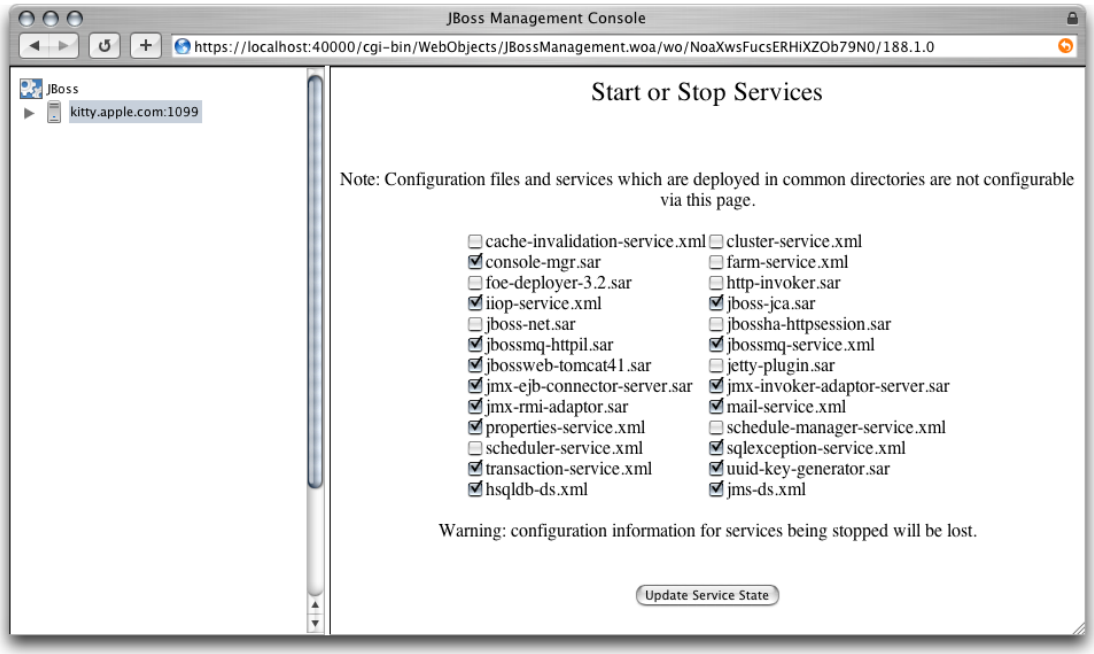


2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Starting and Stopping Services

You can start and stop services while managing or configuring application servers. To do so, in the application server list, select the application server you want to configure, and click Start/Stop Services in the Host Information pane. The Start or Stop Services pane is displayed in the right side of the JBoss Management Console window, as shown in Figure 4-7.

Figure 4-7 The JBoss Management Console window showing the Start or Stop Services pane



Creating a Data Source

You can create a data source while managing or configuring application servers. To do so, select the application server you want to add the data source to in the server list. Next, enter the name of the data source in the Datasource Name text field in the Create a Datasource group in the Host Information pane, choose a data-source type from the Datasource Type pop-up menu, and click Create Datasource.

Enter the appropriate information in the Local TX Datasource pane, and click Update.

> **Note:** Make sure to enter a unique JNDI name in the Jndi Name text field of the Local TX Datasource pane.

The newly added data source appears under the Resources group in the application-server list.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Creating a Topic or a Queue

You can create a topic or a queue while managing or configuring an application server. Follow these steps to create a topic or a queue:

1. In the application server list, select the application server you want to add the topic or queue to.
2. From the Topic or Queue pop-up menu in the Create a Topic or Queue group in the Host Information pane, choose Topic or Queue.
3. In the Topic or Queue Name text field, enter the name of the topic or queue.
4. Enter the name of the file in which the topic or queue configuration is to be saved in the Filename text field.

Deploying Applications

You can deploy applications while managing or configuring application servers. Follow these steps to deploy an application:

1. Select the application server you want to deploy the application or service on in the application server list.
2. In the “Select an application to deploy” group in the Host Information pane, click Choose File, and choose the file to deploy.
3. Click Deploy Application.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Balancing User Load and Replicating Sessions

HTTP load balancing provides a way to distribute user load among a group of application servers. The application servers can be standalone or configured as a cluster, in which case they are know as *nodes* . Load balancing is better used with *sticky sessions* . This means that once the load balancer (a web server) forwards a client request to a particular application server, it sends all further requests from the client to the same application server.

Using load balancing across standalone application servers allows you to scale your deployment with little increase in request-processing overhead. However, when an application server fails, other application servers cannot pick up the failed-server's load, which may provide an undesirable user experience: Users may have to log in to the application again or may lose the contents of their "shopping carts."

Load balancing across clustered application servers (or nodes) provides session replication among the nodes, so that when a node fails, another node can take over its duties with little or no user impact. However, as you add nodes to the cluster, each request may take longer to process.

This chapter explains how to enable an application to be distributable among cluster nodes and walks you through configuring HTTP load balancing for Sun's Pet Store using three computers: One serving only as the web server and load balancer, and the other two serving as application-server nodes.

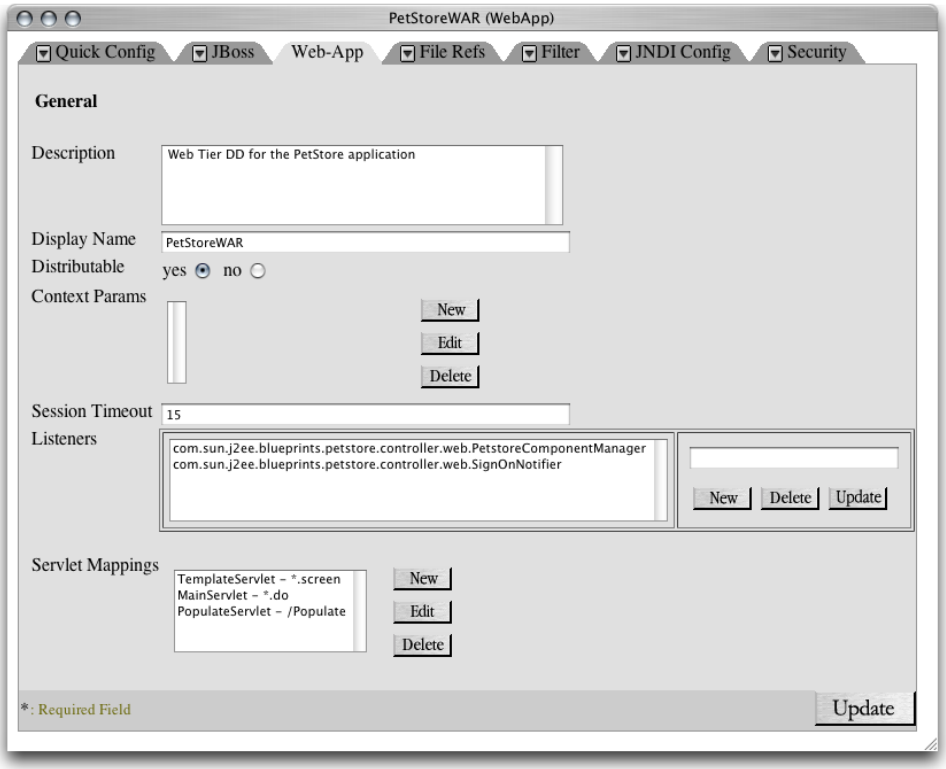
Distributable Applications

Before deploying an application in a cluster of nodes using the `deploy-cluster` configuration, make sure that the application is distributable.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

To make an application distributable set Distributable to yes in the Web-App pane of the application's WebApp window. Figure 5-1 shows the WebApp window of the `petstore.ear` archive.

Figure 5-1 The WebApp window of the `petstore.ear` archive



Load Balancing and Clustering

Load balancing provides a way to distribute user load among application servers. Clustering enables session failover when a node in a cluster becomes unavailable. Load balancing can be used with a group of standalone application servers or with a cluster or application server or nodes. This section describes a simple, three-computer setup in which one computer runs the web server and balances user load among two application servers.

Start by stopping the Web service on the web-server computer and the application servers on the nodes. Then follow the steps described in the following sections.

Enable Load Balancing in the Web Server

Follow these steps to configure a computer as the web server and load balancer for a deployment:

1. Launch Server Admin, if it's not already running.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

2. Select Web in the Computers & Services list, and click Settings in the configuration pane.
3. Click the Modules tab and select "jk_module," which is at the bottom of the modules list.
4. Click the Sites tab.
5. Double-click the appropriate site in the list (by default there's only one), which should be enabled.
6. Click the Options tab, and deselect Performance Cache.
7. Click Save.
8. Open `httpd.conf` file, located in `/etc/httpd`, in a text editor.
9. Look for `<IfModule mod_jk.c>`.
10. Add `JKMount /petstore/* loadbalancer` as the last item of the `IfModule` element.

    The `IfModule` element should look similar to this:

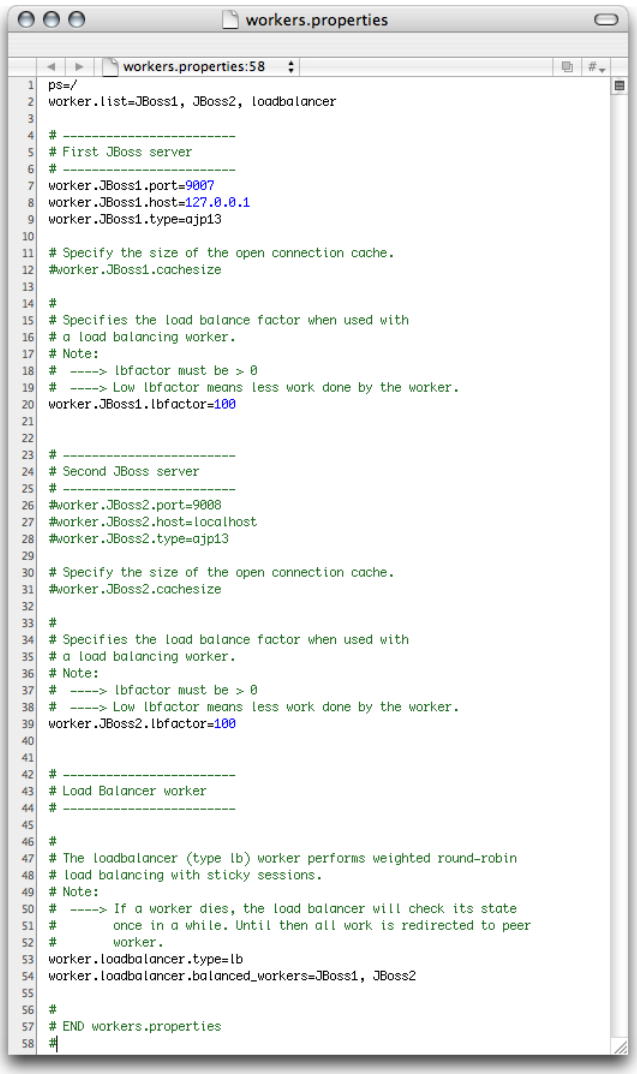
```
<IfModule mod_jk.c>
        JKWorkersFile /etc/httpd/workers.properties
        JKLogFile /var/log/httpd/mod_jk.log
        JKLogLevel error
        JKMount /*.jsp JBoss1
        JKMount /servlet/* JBoss1
        JKMount /examples/* JBoss1
        JKMount /petstore/* loadbalancer
</IfModule>
```

11. Save the file.
12. Open the `workers.properties` file, which is also located in `/etc/httpd`, in a text editor.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

The file as configured in OS X Server is shown in Figure 5-2.

Figure 5-2 The `workers.properties` file in `/etc/httpd`



13. Change line 2 to:

```
worker.list=loadbalancer
```

14. Change line 8 so that it references the first node. It should look similar to this:

```
worker.JBoss1.host=node1.mydomain.com
```

15. Change line 12 to:

Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

```
worker.JBoss1.cachesize=10
```

16. Uncomment lines 26 through 28.
17. Change line 26 so that it looks like this:

```
worker.JBoss2.port=9007
```

18. Change line 27 so that it references the second node. It should look similar to this:

```
worker.JBoss2.host=node2.mydomain.com
```

19. Change line 31 to:

```
worker.JBoss2.cachesize=10
```

20. Add the following line to the file to enable sticky sessions:

```
worker.loadbalancer.sticky_session=1
```

21. Save the file.

Enable Load Balancing in the Application Servers

For load balancing to work, each application server has to report its existence to the web server. Follow these steps to configure the application-server so that they identify themselves to the web server:

1. Open the `jboss-service.xml` file, located at `/Library/JBoss/3.2/deploy-cluster/deploy/jbossweb-tomcat41.sar/META-INF`, in a text editor.

   For non-cluster deployment, open the `jboss-service.xml` file at `/Library/JBoss/3.2/deploy-standalone/deploy/jbossweb-tomcat41.sar/META-INF`.

2. Look for the following line:

```
<Engine name="MainEngine" defaultHost="localhost">
```

3. Edit the line so that it looks like this:

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

```
<Engine jvmRoute="JBoss1" name="MainEngine" defaultHost="localhost">
```

4. Look for the following lines:

```
<!--Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
    port="9007" minProcessors="5" maxProcessors="200" address="0.0.0.0"
    enableLookups="false" acceptCount="50" debug="0"
    connectionTimeout="20000"
    protocolHandlerClassName="org.apache.jk.server.JkCoyoteHandler"/-->
```

5. Remove the `!--` at the beginning of the first line and the `--` and end of the last line while making sure to leave the left angle bracket and the right angle bracket in place.
6. Save the file.
7. Repeat steps 1 through 6 for the second application server, but set `jvmRoute` to `"JBoss2"` in step 3.

Test the Configuration

Follow these steps to make sure that client requests are balanced among the application servers:

1. Start the Web service in the web-server computer.
2. Start the application server in each of the application-server computers and run the following commands on both:

```
$ cd /Library/JBoss/Logs
$ tail -f localhost_access<today's_date_YYYY-MM-DD>.log
```

3. In the web-server computer, connect to `http://<host_name>/petstore/index.jsp`. The first node should show a log entry similar to this:

```
17.203.255.255 - - [26/Sep/2003:15:56:58 -0800] "GET /petstore/index.jsp
 HTTP/1.1" 200 2769
```

4. Now, access the same URL from another computer. You should see a log entry in the second node.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Document Revision History

This table describes the changes to *Java Application Server Guide* .

Date	Notes
2005-08-11	Changed the title from "OS X Server Java Application Server Administration." Made minor edits throughout.
2003-10-22	First version.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Glossary

application server JBoss instance, which is started through Server Admin.

CMP (container-managed persistence) Enterprise bean persistence model in which the J2EE container is responsible for persisting enterprise-bean instances to a data store and populating the fields of enterprise-bean instances when they are retrieved.

deployment tool HTML-based application through which J2EE application or component archives can be configured or assembled in preparation for deployment in OS X Server.

EJB (Enterprise JavaBeans) Specification that provides an infrastructure through which data-based components can be developed and deployed in a variety of platforms.

J2EE (Java 2, Enterprise Edition) Specification that defines a platform for the development and deployment of Web applications. It describes an environment under which enterprise beans, servlets, and JSP pages can share resources and work together.

JBoss Java-based open-source application server capable of deploying J2EE-based applications. JBoss provides many useful features in addition those defined in the J2EE standard, including support for clustering, session replication, mail, and security.

JMS (Java Message Service) Java-based programming interface that implements an asynchronous message-exchange system. It facilitates the development of message-based applications. JMS is part of the J2EE platform.

management tool HTML-based application through which an application-server configuration can be modified. It also allows for the viewing of statistics of resources and services deployed on application servers, starting and stopping services, and adding topics, queues, and data sources.

Pet Store Pet Store is a sample J2EE application from Sun Microsystems, which showcases the power and flexibility of the J2EE platform.

queue A queue is a JMS construct that allows for point-to-point messaging between applications. A message sent to a queue can be received by only one application. When several applications are subscribed to the queue, the messages are load balanced between the subscribers.

server Computer running OS X Server.

topic Topics are one of the message distribution center types for J2EE-based applications. Message senders send messages only to topics instead of specific applications, while only the applications interested in receiving messages sent to a particular topic subscribe to the topic and, therefore, receive the messages sent to it. A topic can have one or more subscribers. Any message sent to the topic is broadcasted to all the topic's subscribers.

2005-08-11 | Copyright © 2003, 2005 Apple Computer, Inc. All Rights Reserved.

Apple Inc.
Copyright © 2003, 2005 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer or device for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-branded products.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, Finder, Logic, Mac, NetInfo, and OS X are trademarks of Apple Inc., registered in the U.S. and other countries.

Java is a registered trademark of Oracle and/or its affiliates.

APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT, ERROR OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

Some jurisdictions do not allow the exclusion of implied warranties or liability, so the above exclusion may not apply to you.